SPAWAR
Systems Center
San Diego

TECHNICAL DOCUMENT 3176
February 2004

# SSC San Diego REMUS UUV Sensor Integration

Vladimir Djapic
Rich Arrieta
Brian Granger
Andy Dreiling



SSC San Diego

TECHNICAL DOCUMENT 3176
February 2004

# SSC San Diego REMUS UUV Sensor Integration

Vladimir Djapic
Rich Arrieta
Brian Granger
Andy Dreiling



SPAWAR
Systems Center
San Diego

SSC San Diego
San Diego, CA 92152-5001

**SSC SAN DIEGO**
**San Diego, California 92152-5001**

**T. V. Flynn, CAPT, USN**
**Commanding Officer**

**R. F. Smith**
**Executive Director**

## ADMINISTRATIVE INFORMATION

The work described in this report was performed in the Ocean Systems Division (Code 274) of the Intelligence, Surveillance, and Reconnaissance Department (Code 270) of Space and Naval Warfare Systems Center, San Diego (SSC San Diego).

Released by
Rich Arrieta, Head
Marine Systems Branch

Under authority of
M. N. Westmoreland, Deputy for Operations
Ocean Systems Division



### Acknowledgments

This report is the product of an extensive collaboration between SSC San Diego; Woods Hole Oceanographic Institution; University of California, Riverside; and Nomadics, Inc. The authors would particularly like to thank Roger Stokey of Woods Hole Oceanographic Institution and Matthew Dock of Nomadics, Inc. Following an extensive collaboration effort, Roger Stokey authored Appendix A, REMUS Generic NMEA183 Interface. Matt Dock provided critical input to defining the generic message format and was involved with all aspects of the mechanical and electrical integration process. We would also like to thank Dr. Jay Farrell of University of California, Riverside, for his insight and contributions to the development of the generic message structure and overall communications protocol.

# CONTENTS

# 1. INTRODUCTION

## 1.1 PURPOSE

This document specifies a standard for integrating environmental sensors onto the Space and Naval Warfare Systems Center, San Diego (SSC San Diego) Remote Environmental Monitoring Unit Support (REMUS) unmanned underwater vehicle (UUV). If sensor developers construct sensors in accordance with the guidance provided in this document, they can do so without having access to the vehicle itself, and final electrical and mechanical integration of the unit will be a trivial task.

## 1.2 SENSOR CATEGORIES

Sensors designed for integration onto the REMUS UUV may come in a variety of sizes and shapes. Thus, three sensor category descriptions are provided in the following sections.

### 1.2.1 Category One Sensors

Category One includes sensors that require roughly the same space as the Optical Backscatter (OBS) (7 inches long and 1.25 inches in the diameter) or SeaPoint Fluorometer (6.6 inches long and 2.5 inches in the diameter).

These sensors are small enough such that the pressure housing can be designed to fit into the existing OBS or fluorometer nosecones respectively (a custom mounting bracket may be required), as shown in Figures 1 and 2. This configuration would require no additional modifications to the REMUS UUV.

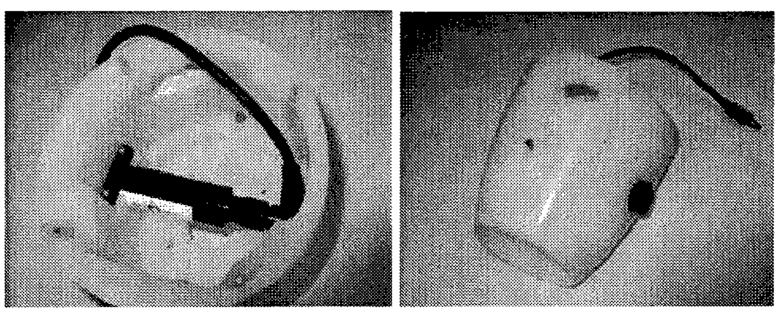


Figure 1. Optical Backscatter Sensor (OBU) mount.

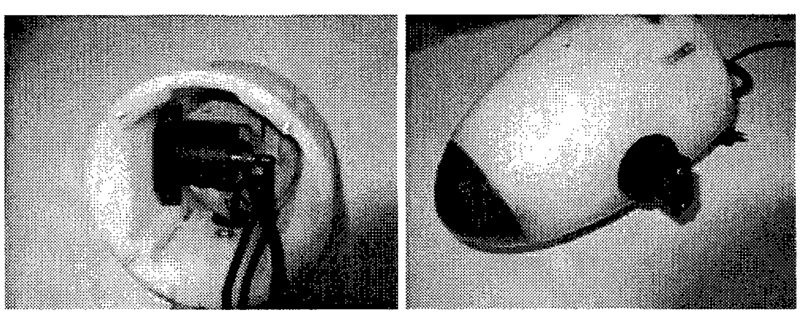


Figure 2. Fluorometer mount.

### 1.2.2 Category Two Sensors

Category Two sensors are as big as or slightly larger than the fluorometer or the OBS and may require custom mounting into a "dummy" nosecone and/or may require some additional space, but not enough to justify making a separate vehicle module. These sensors will be mounted into a "dummy" nosecone and will be mated to the vehicle via an intermediary collar device designed and provided by SSC San Diego. The collar will be the same diameter as the REMUS (7.5 inches) and will bolt to the existing holes on the front bulkhead of the vehicle's Acoustic Doppler Current Profiler (ADCP) unit. Specifications for the "collar" are provided in Section 2.1. Specifications for a standard "dummy" nosecone are provided in Section 2.5. Figure 3 shows the REMUS UUV with the collar and nosecone installed.

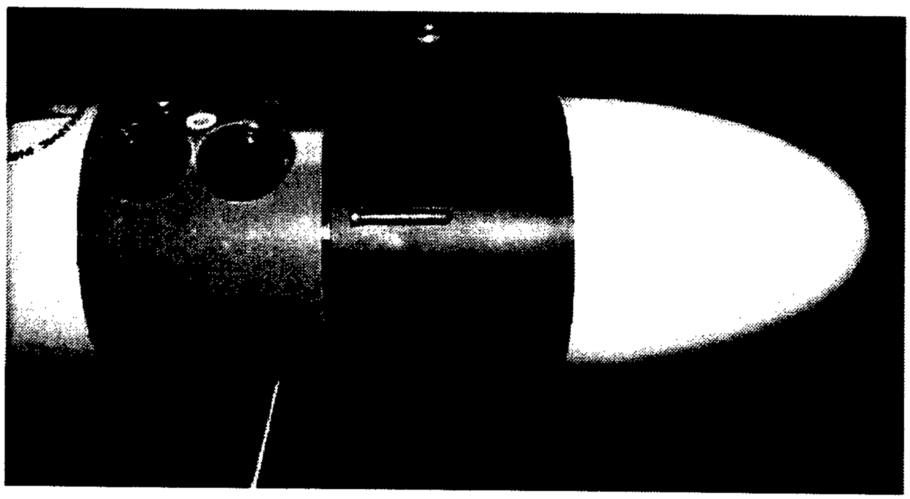


Figure 3. REMUS UUV with collar and "dummy" nosecone installed.

Since the nosecone is now removed from the vehicle's forward bulkhead, the collar will accommodate the LBL (long baseline) transducer, serial and network connections, and the current, temperature, depth (CTD) sensor. The dummy nosecone thus can be used entirely for sensor hardware. The sensor may also overflow into additional space afforded by the collar device. While the collar measures 6.25 inches in length, the bulkhead connectors extend 2.25 inches into the collar, and the CTD sensor extends 4.25 inches into the collar. So while it does afford some extra space, a designer would have to work around existing equipment.

This configuration will likely not include (but could) the USBL (ultra short baseline) transducer by including a USBL mount on the dummy nosecone and routing a USBL through the nosecone and the collar.

### 1.2.3 Category Three Sensors

Category Three sensors require enough space (volume) to justify an additional vehicle module. Integration of such sensors will follow a modular approach by placing additional body sections between the ADCP module and the nosecone. Again, the collar device will mount to the ADCP forward bulkhead and accommodate the LBL, serial and network connector access, and a CTD flow port. A "dummy" nosecone will mount to the front of the new module. This configuration will not likely include a USBL transducer (unless the USBL cable is somehow routed through or around the

module). Figure 4 shows the SSC San Diego REMUS with Nomadics' SeaPup explosive sensor installed. Forward of the ADCP is the collar. It is difficult to identify because both are black, but ahead of the collar (with the sensor protruding from the side) is the explosive sensor. Forward of the sensor is the "dummy" nosecone.

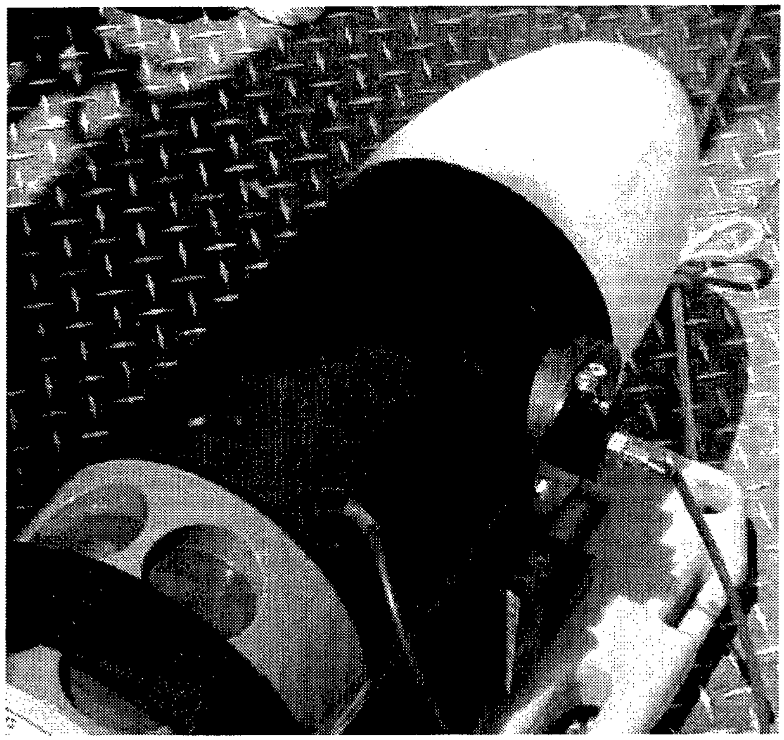


Figure 4. SSC San Diego UUV with "SeaPup" explosive sensor installed.

A pressure vessel design for a new vehicle module is specified in Appendix B. It will bolt to the collar device using the same bolt pattern specified in Section 2.1 and be capped with a "dummy" nosecone (Section 2.5). These sensors may be fully contained within the pressure vessel or may use the flooded space of the collar and dummy nosecone for sensor mounting or some pressure-insensitive hardware. Specifics for the mechanical integration for Category Three sensors are provided in Section 2.

The modular design used for new vehicle module integration allows multiple sensors to be integrated onto the vehicle simultaneously. This concept is further explained in Section 2.

# 2. MECHANICAL INTERFACE

As mentioned in Section 1.2, sensor packages similar in size and shape to the SeaPoint Fluorometer or OBS may be mounted in existing nosecones (Category 1). If a small additional volume is required, this may be accommodated by the addition of the collar device and a standard or customized "dummy" nosecone (Category 2). Sensor packages larger than this (Category 3) will need to be housed in a new body section forward of the ADCP module. This new section will be mounted to a collar device provided by SSC San Diego. With this modular design, multiple sensors can be integrated onto the REMUS UUV. Figure 5 shows a conceptual view of the modular, multiple-sensor integration design.

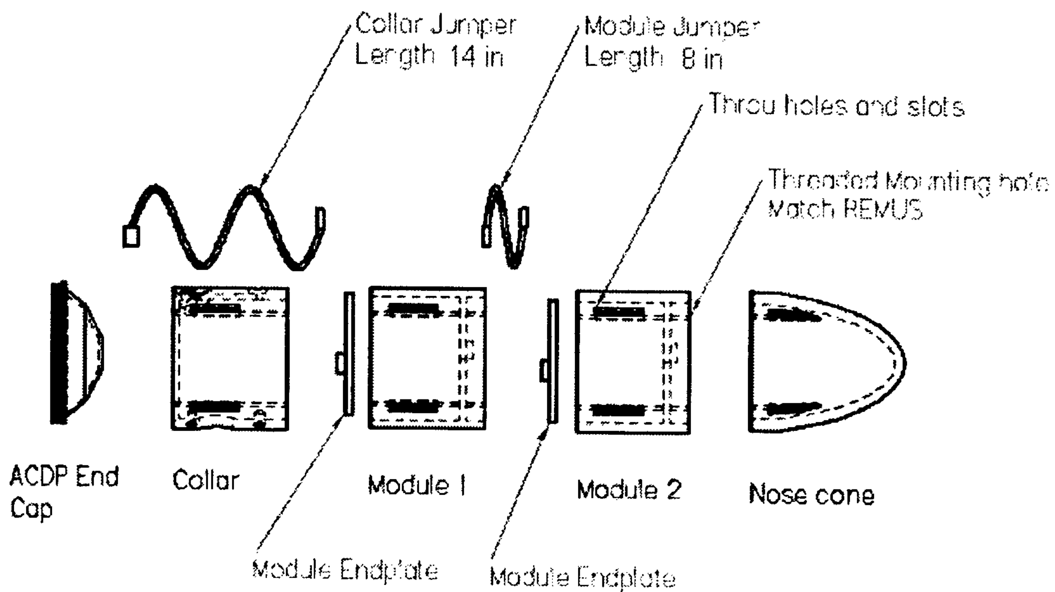



Figure 5. Modular sensor design.

## 2.1 COLLAR

SSC San Diego will provide a collar that will attach to the REMUS forward of the ADCP module. This collar will be the same diameter as the main body of the vehicle (7.5 inches) and will be 6 inches long. The LBL transducer will be relocated from the nosecone to the collar. This will allow the LBL transducer to be connected without the need for any type of extension cable. Unless required for a particular mission, the USBL transducer will be left out of the system. This will eliminate the need to string cables through or around sensor packages.

The collar will be attached to the same four attachment points that secure the nosecone to the vehicle.

The same bolt pattern will be duplicated on the front end of the collar for attachment of a sensor module or custom nosecone. Mounting holes are included for installation of the LBL transducer, a water flow port for the RTD (resistance temperature detector) sensor, and access holes for the serial and Ethernet connections. Figure 6 provides a three-dimensional view of the SSC San Diego collar device, and Figure 7 provides specifications for the bolt pattern and required fasteners.

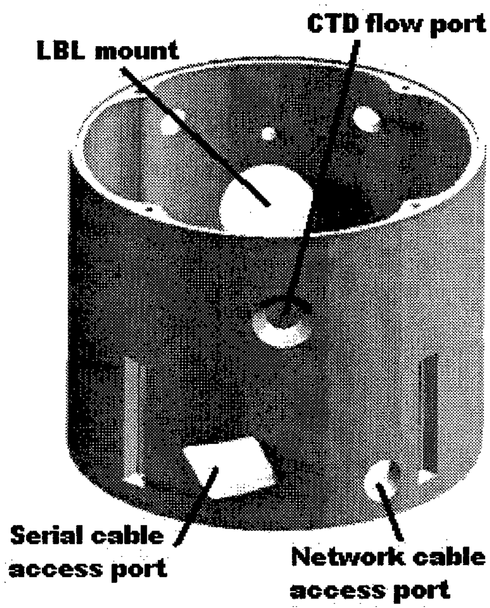



Figure 6. SSC San Diego collar device.

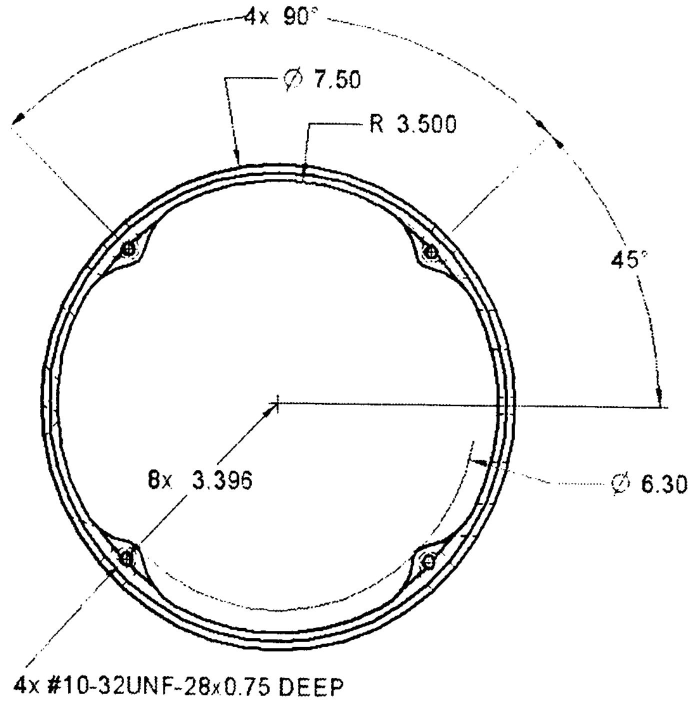



Figure 7. Bolt pattern for collar and sensor modules.

## 2.2 CABLING

### 2.2.1 Collar Jumper Cable

The collar jumper is 14 inches long. It has an 8-conductor, female-socket connector (Impulse LPMIL-8-FS) for connection to the ADCP bulkhead, and an 8-conductor, male-pin connector (Impulse LPMIL-8-MP) for connection to the first sensor module. The jumper will extend from the ADCP bulkhead, through the collar, and attach to the first sensor module as shown in Figure 5. At 14 inches long, it will extend approximately 8 inches beyond the front edge of the collar.

### 2.2.2 Module Jumper Cable

If more than one sensor module is installed, a module jumper cable will be required to connect them (daisy-chain configuration). This cable will consist of two 8-conductor, male-pin connectors (Impulse LPMIL-8-MP) molded on either end of an 8-inch cable. Each end of a sensor module will have an 8-conductor, female-socket, bulkhead connector (Impulse LPMBH-8-FS) for receiving and transmitting sensor power and data. The Module Jumper cable is also shown in Figure 5.

## 2.3 SENSOR MODULE

The Sensor Module will include a pressure vessel (see Section 2.4), which may or may not contain the sensor. Each module must be designed to attach to the SSC San Diego collar (or to another sensor module) using the standard four-bolt pattern found on the front bulkhead of the ADCP as specified in Figure 7. They must also provide four threaded holes in the same pattern on the forward end for

attachment of the nosecone or another sensor module.

The forward end of the sensor module will include a 1-inch space to provide clearance for the module jumper cable, the connector, and the connector on the next module. The Impulse LPMBH bulkhead connectors stand approximately 0.63 inches above the bulkhead surfaces. With proper placement of the connectors (described below), 1-inch clearance should be sufficient between modules. The aft bulkhead of the module should be flush with the end of the module body. Connector orientation is shown in Figure 8.

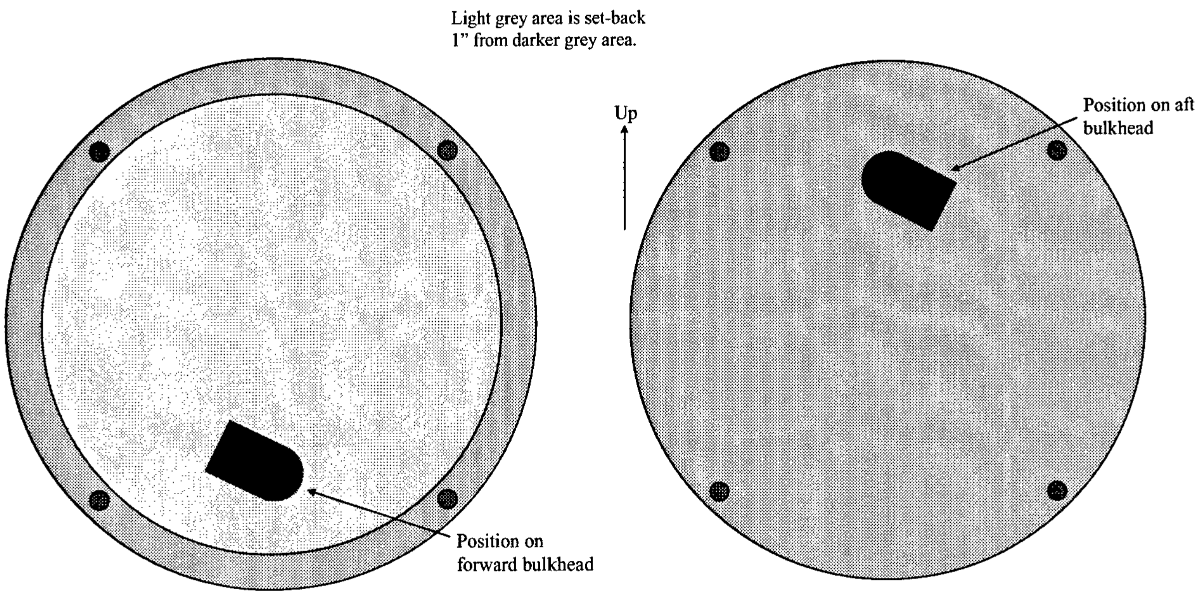



Figure 8. Sensor module bulkhead connectors.

The bulkhead connectors must be located such that they will not contact each other when sensor modules are connected together. The standard for connector placement will be as follows: for the aft bulkhead of a sensor module, the connector should be mounted at least 2 inches above the center of the bulkhead face, and the open face of the connector should be oriented to the 4 o'clock position. For the forward bulkhead, the connector should be mounted at least 2 inches below center, and should be oriented to the 10 o'clock position. Orientation of the connector assumes you are looking at the respective face. See Figure 8.

Sensor modules that are at least partially flooded must also have enough holes in the main body to ensure that they can flood/drain within 5 to 10 seconds. Holes must also be drilled near the forward end of the module so that the connector/cable space (between the modules) can easily flood/drain.

### 2.3.1 Buoyancy Considerations

All sensor modules must incorporate some sort of variable buoyancy. The REMUS vehicle is positively ballasted by the manufacturer to approximately 1 lb. (± 0.25 lb.) for operating in most ocean environments. A small amount of space is available in the collar and in the nosecone. Foam or weights can be added to these spaces to compensate for the buoyancy (positive or negative) of the

sensor module.

Any buoyancy material added to a module must be able to tolerate a depth of 100 m without crushing or absorbing water. One type of foam that has been used successfully is Syntactic Foam made by Flotation Technologies rated to 100 m depth (www.flotec.com). If ballast weights are necessary, they should be added to the bottom of the sensor module. The REMUS design is shown in Figure 9. The weight bar pictured is located on the underside of the vehicle. Weights of various sizes slide onto the bar. The weights are prevented from falling vertically by the bar itself, and are prevented from sliding horizontally by the installation of a setscrew. Since penetrations into the pressure vessel are not desired, the weight bar is held in place by an adhesive product such as 3M 5200 Fast Cure.

The sensor modules must also maintain a vertical orientation between center of mass (CM) and center of buoyancy (CB), with CM located below CB. Any large imbalance could give the vehicle a tendency to roll to one side.

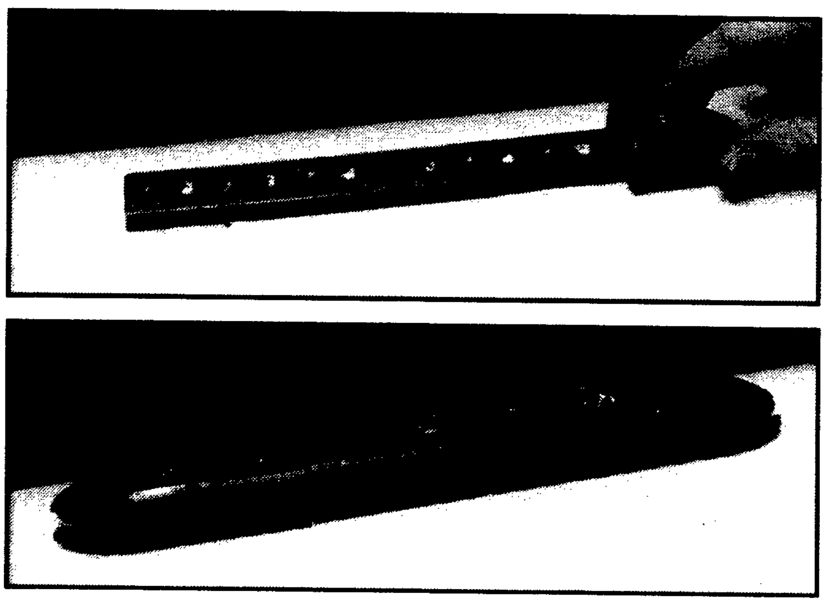


Figure 9. Variable ballast onboard REMUS 100 vehicle.

## 2.4 PRESSURE VESSEL

SSC San Diego is currently developing a pressure vessel based on the standard sensor module shell. Preliminary designs for the basic shell and pressure vessel are found in Appendix B. The design drawings will be made available to other principal investigators involved in designing sensors for the REMUS vehicle.

## 2.5 NOSECONE

A hollow "dummy" nosecone will be bolted to the end of any new body section(s). The space inside the nosecone will have attachment points for adding flotation or weight in order to balance the modified vehicle. If necessary, the nosecone can be customized to house sensors or parts of sensors. See Figure 10.

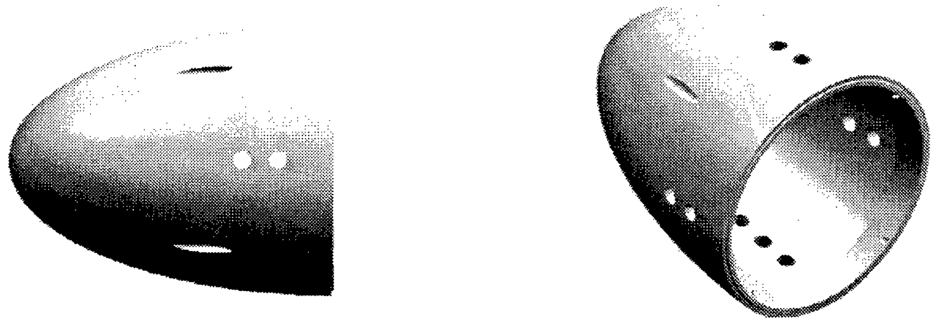


Figure 10. "Dummy" nosecone.

As previously mentioned, for smaller sensor packages that are comparable in size to the SeaPoint Rhodamine Fluorometer it may be possible to integrate the package into a custom nosecone assembly, or into the stock nosecone made for the Fluorometer or OBS sensor (a custom mounting bracket would likely need to be made). Detailed specifications for the "dummy" nosecone are found in Appendix C.

# 3. ELECTRICAL AND COMMUNICATIONS INTERFACE

## 3.1 SENSOR CONNECTOR

Any new sensor will make use of the sensor input interface provided on the REMUS for the OBS and Fluorometer sensors. The standard sensor-interface connector on the REMUS is an 8-pin, bulkhead connector manufactured by Impulse Enterprise, Inc., San Diego, CA; Part number LPMBH-8-MP (See Figure 11). The details of each pin are listed in Table 1. Additional specifications for the electrical connector are provided in Section 3.2.1.

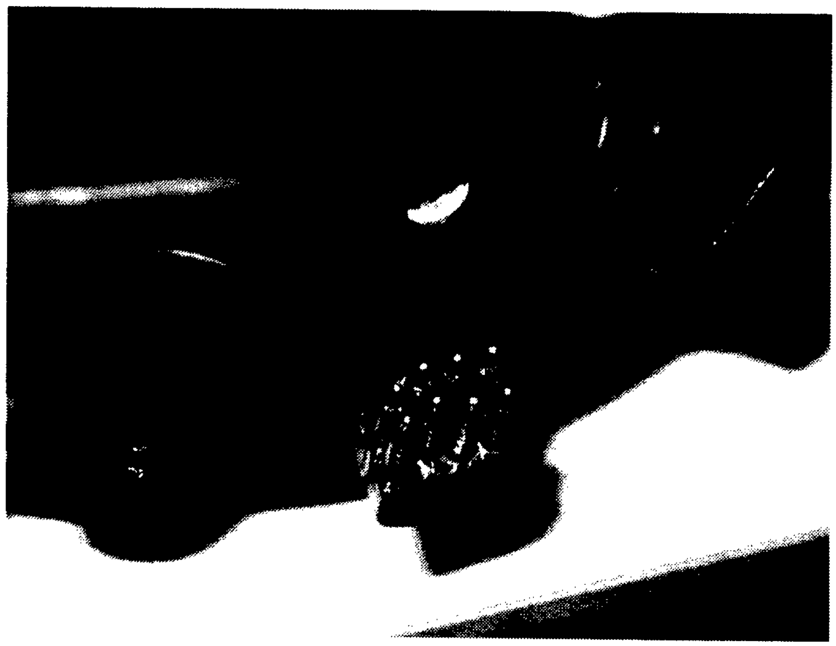


Figure 11. Impulse bulkhead connector for modular sensors.

Table 1. Pin functions on impulse connector.

| Pin Number | Function |
|---|---|
| 1 | Power Common |
| 2 | +12 VDC (30 W max) |
| 3 | Gain Control A |
| 4 | Analog Signal Input (0 to 5 VDC) |
| 5 | Analog Signal Common |
| 6 | Gain Control B |
| 7 | RS-232 TX (115,200 baud max) |
| 8 | RS-232 RX (115,200 baud max) |

The Rhodamine fluorometer sensor uses pins 1 through 6, while pins 7 and 8 are reserved for sensors with serial (RS-232) outputs. Pins 3 and 6 are used to set the gain control for the fluorometer. Depending on the settings for the vehicle .INI file, these two pins will be set to 0 or +5 VDC. The combination of the two will determine four possible gain control settings.

The fluorometer output voltage (between 0 and 5 VDC) is sensed by pins 4 and 5. This analog signal is then sampled by a built-in 24-bit analog to digital converter. It is saved (after gain and offset corrections) as part of the normal REMUS telemetry stream at 1 Hz. It is possible to store the raw (0 to 5 VDC) values from the analog input line in an ASCII file at up to 9 Hz.

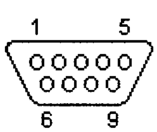
The REMUS can provide a maximum of 30 W of power (at 12VDC) to pin 1. In its normal configuration, the vehicle consumes between 45 and 50 W with the speed set at 3 knots. At this rate, the batteries will typically last about 20 hours. At 5 knots, power consumption jumps to around 105 W and the battery life drops to about 9 hours. These estimates can be used to approximate the impact of a given sensor package on vehicle battery life.

## 3.2 ADVANCED SENSOR COMMUNICATIONS

### 3.2.1 Serial Communication Basics

Serial communication is the most common low-level protocol for communicating between two or more devices. Normally, one device is a computer, while the other device can be a modem, a printer, another computer, or a scientific instrument such as an oscilloscope or a function generator. In our case, it is an external sensor. As the name suggests, the serial port sends and receives bytes of information in a serial fashion — one bit at a time. These bytes are transmitted using either a binary format or a text (ASCII) format. Over the years, several serial port interface standards for connecting computers to peripheral devices have been developed. These standards include RS-232, RS-422, and RS-485 — all of which are supported by the serial port object. Of these, the most widely used standard is RS-232 (Recommended Standard number 232).

The standard serial port connector for every computer is a 9-pin connector. The pins are arranged in standard fashion. The connector has five pins on the top, and four on the bottom, with the first pin being the top left pin. The pin-out is show in Figure 12.




| DB-9M | Function | Abbreviation |
|---|---|---|
| Pin #1 | Data Carrier Detect | CD |
| Pin #2 | Receive Data | RD or RX or RXD |
| Pin #3 | Transmitted Data | TD or TX or TXD |
| Pin #4 | Data Terminal Ready | DTR |
| Pin #5 | Signal Ground | GND |
| Pin #6 | Data Set Ready | DSR |
| Pin #7 | Request to Send | RTS |
| Pin #8 | Clear to Send | CTS |
| Pin #9 | Ring Indicator | RI |

Figure 12. RS-232 DB-9 connector.

### 3.2.2 Serial Communications on the REMUS UUV

Thus far, sensors integrated onto the REMUS UUV have used just 3 of the 9 pins on the serial connector: pin 2 for Received Data, pin 3 for Transmitted Data, and pin 5 for Signal Ground.

Pin 7 (on the Impulse 8-pin, bulkhead connector) is reserved for serial transmission of the data from the sensor to the REMUS, while pin 8 is reserved for data that is serially sent from the REMUS main computer to the sensor. All data is sent in strings of ASCII characters at baud rates of up to 115,200. Multiple sensors can potentially make use of the same RS-232 port by multiplexing the individual data streams and separating them during post-processing.

### 3.2.3 Generic Sensor Communication Protocol

The REMUS Generic NMEA 183 interface is designed to allow any instrument great flexibility in creating its own message specification. The complete NMEA 183 Interface standard for the REMUS UUV is provided in Appendix A, but a brief description will be provided here.

Sensor messages will be transmitted to REMUS, which will parse and log them. In order to use sensors that output serial data, a generic .DLL file is added to the standard REMUS VIP (Vehicle Interface Program) software so that the data will be properly stored. This will also allow the data to be exported as comma-delimited text or as a MATLAB data file.

Each message type is logged within the REMUS telemetry file every second. A sensor with a sampling rate higher than this will also be accommodated. The full data set is recorded in the GENERIC.DAT file on the vehicle.

An example message format is provided in Table 2. Up to 12 messages can be sent, and each one can have up to eight floating type data values. Therefore, it is possible to accommodate multiple sensors onboard REMUS simultaneously. The messages from the multiple sensors are multiplexed and sent via the same serial port. A unique five-character header (prefixed by a $ sign) will start each message. A checksum, line-feed, and carriage return will follow the data values.

Table 2. Schematic that describes messages sent from the sensor to REMUS.

| | Five character unique header | | Up to 8 data values | | | |
|---|---|---|---|---|---|---|
| 12 Messages | $_____ | float | float | float | float | checksum |
| | $_____ | float | float | float | float | checksum |

If any variable is prefixed by an exclamation point ('!') or the "at" sign ('@'), REMUS will indicate that the instrument is faulted. A sensor manufacturer can place a pressure sensor inside of the module and if, for instance, pressure drops below a certain threshold indicating loss of vessel integrity, the sensor can send a '!' or a an '@' in any of the messages. REMUS will then execute the appropriate abort procedures.

The message descriptor will be specified in the vehicle ".ini" file. The format is provided in Table 3. Other sensor parameters (such as baud rate for sensor data transmission) will be specified here as well.

Table 3. Generic Instrument paragraph in REMUS .ini file.

```
[Generic Instrument]
Instrument present=yes
Instrument name=NMSU
Abort launch if broken=yes
Baudrate=19200
Message description=$NMSUE, max_value
Max disk usage (Megs)=20
Send at startup=
Send at launch=
Send at mission over=
Max disk usage (Megs)=20
Disable USBL array=yes
Message description=#$PQRXT,variable_name,variable_name,...
```

In the specific case illustrated in Table 3, the instrument present is the "NMSU" sensor. The vehicle is programmed to continue launch of the sensor if it is "broken," but could be configured such that it would not. The communication baud rate is set at 19200 and the REMUS vehicle expects one variable named "max_value." The vehicle is capable of sending commands to the sensor (i.e., start pump, turn-on camera, etc.) at startup, launch, and mission over. Users can set the limit of disk space used for sensor data. In this case, the USBL transponder is disabled, and finally the message description is provided. Again, the entire content of the Sensor Communication Protocol can be found in Appendix A.

# 4. POINTS OF CONTACT

Rich Arrieta
Project Elena Program Manager
arrieta@spawar.navy.mil
619.553.1968

Brian Granger
Project Engineer
bgranger@spawar.navy.mil
619.553.5275

Vladimir Djapic
Technical Lead
djapic@spawar.navy.mil
619.553.1654

Andy Dreiling
Technical Lead
dreiling@spawar.navy.mil
619.553.4033

## APPENDIX A: COMMUNICATIONS PROTOCOL

# REMUS Generic NMEA183 Interface

June 24, 2003

### Applicability

This information is applicable for vehicle, host, and .DLL software releases dated on or after March 17, 2003, and supercedes any prior information.

### Overview

The REMUS NMEA 183 Instrument interface is designed to allow any instrument using this protocol to operate as part of REMUS system. It allows the instrument designer great flexibility in creation of his/her own message specification and to transmit these messages to REMUS, which will parse and log them.

A standard NMEA183 message is an ASCII message that begins with a '$' sign and a 5 character message type, and ends with an asterisk ('*'), followed by a 2 digit hexadecimal checksum, and then a carriage return-linefeed pair. The REMUS NMEA specification allows up to 8 floating-point values as parameters to the message. Up to 12 different messages may be specified, thus up to 96 different variables can be monitored from a single instrument.

One of each message type specified is logged within the REMUS telemetry (.RLF ) file per second. Data that is transmitted at a rate faster than that is down sampled. Data may also be optionally logged to a separate "GENERIC.DAT" file. There is no such restriction on the data rate of messages logged to this file. All of the data is logged to this file.

### System Configuration

The system adds the entries to the vehicle configuration file (remus_v.ini) in its own section:

**[Generic Instrument]**
**Instrument present=YES**

Setting this entry to "no" allows the instrument to be removed from the system. This will generate a warning message at startup; however, other than that operation is as if the instrument were not present in the system.

**Instrument name=Sea Dog**

This entry "names" the instrument. Up to 11 ASCII characters may be used. This name is used on the annunciator (idiot light) for the instrument on the REMUS VIP, as well as in error messages, etc., regarding the instrument.

**Disable USBL array=NO**

This parameter should be set to YES if the instrument installation requires removal of the USBL

array. It prevents the vehicle from trying to use a non-existent array for navigation. Obviously vehicles without USBL capability will not have this line.

**Abort launch if broken=YES**

By setting the entry to no, the vehicle can be launched even if the instrument is broken or otherwise malfunctioning. This should only be done when instruments data is not critical to the mission, since it is unlikely that valid data will be collected.

**Max disk usage (Megs)=10**

This entry limits the maximum amount of disk usage by the instrument. Because the instrument could generate a huge amount of data, this setting can be used to limit the total quantity. Note that interface will automatically generate a warning message at startup if the instruments data file is already larger than 1 megabyte.

**Baudrate=9600**

This sets the baudrate for communication with the instrument. Any baudrate up to and including 115200 is supported.

**Send at startup=$PTEST, Send this at start**
**Send at startup=**
**Send at launch=$PTEST, Launched**
**Send at launch=**
**Send at mission over=$PTEST, ITS ALL OVER**
**Send at mission over=**

These messages are transmitted to the instrument at the times indicated. There may be more than one message. The system automatically appends the asterisk, checksum, and carriage return linefeed to the transmission. These messages can be used to set the configuration of an instrument, and/or to enable or disable hardware on the instrument itself; for example, to turn a pump on at launch, and turn it off when the mission is over.

**Message description=$P1MSG,Var00,Var01**
**Message description=$P2MSG,Var10**
**Message description=$P3MSG,Var20,Var21,Var22**
**Message description=#$PQRXT,var_name,var_name, ...**

These entries describe the messages that the instrument will transmit. The first parameter is the NMEA message type. This must begin with a $ sign, and end with a five-character message descriptor. Following that are the names of the variables that the instrument is transmitting. This name can be up to 11 characters long. There should not be any spaces between the comma and the start of the variable name; otherwise the variable name will begin with a space as well. In the above example, REMUS is told to expect three NMEA message types, "P1MSG", "P2MSG", and "P3MSG". The first will have two variables (using the imaginative names "Var00" and "Var01". The second will have only one variable, and the third will have three. The last message description is just a template (reminder) of the format to use for entering the information. REMUS can handle up to 12 different message types. Information about variable names is stored as part of the vehicle telemetry file, so it is available on playback.

## Message Format

Each message from the instrument must begin with a '$', and end with an asterisk followed by a two digit hexadecimal checksum, then an optional carriage return ('\r') and a required linefeed ('\n'). The checksum is calculated by taking a logical exclusive-OR operation of the 8-bit message characters. The checksum excludes the leading '$', checksum delimiter '*' and the checksum itself. The following code fragment can be used to calculate the checksum, and append the required characters to the end of a message:

```
void Write(char *msg)
{
char buffer[200];
char *ptr;
char checksum;
checksum = 0;
ptr = msg+1;
while(*ptr)
checksum ^= *ptr++;
sprintf(buffer, "%s*%02X\r\n", msg, checksum);
...
// add code here to write message to serial port.
}
```

## Serial Debugger

The serial debugger may be used to view data transmitted by the instrument to the vehicle; however it will only display valid message types that it "knows" about and that have a proper checksum. The data is down sampled as required; only one message per type per second is displayed. Typical output from the debugger is as follows:

```
>$P1MSG,123.4,213.4*38
>$P2MSG,233.5*3E
>$P3MSG,234,788.4,00567*3A
>$P1MSG,123.4,213.4*38
>$P2MSG,233.5*3E
>$P3MSG,234,788.4,00567*3A
```

Observe that P1MSG has two parameters, P2MSG has one parameter, and P3MSG has three parameters; this is as described in the message descriptions in the vehicle .INI file (above).

The serial debugger can be used to send commands to the instrument. The command should begin with a '$', and include everything up to, but NOT including, the trailing asterisk, checksum, and cr-lf combination. Those will be added automatically by the vehicle.

## Flags

If an instrument transmits the value of a variable prefixed by an exclamation point ('!'), REMUS will indicate the instrument is faulted, and turn the annunciator red or yellow, depending on how the operator has configured the vehicle (see "Abort launch if broken," above). A fault message is also generated. This can be used to embed diagnostic information within the instrument. The vehicle does not interpret the values sent as being either "good" or "bad." If the value "342" is sent, that will be viewed as good data. If in the next message, "!342" is sent, it will be interpreted as bad.

If the value of a variable is prefixed by an '@', then the mission is immediately aborted. This should only be used in dire, catastrophic emergencies, such as a leak. This event is logged, and the appropriate party is blamed. The upper left annunciator will turn red, and will report "Abort Sig," to indicate that the vehicle was aborted by an external signal. Note that setting "Abort launch if broken" to NO will not prevent the @ sign from aborting the vehicle.

Below are sample messages with the abort and faulted flags set and highlighted for clarity:

```
$P1MSG,123.4,@213.4*78
$P2MSG,233.5*3E
$P3MSG,234,!788.4,00567*1B
```

*Important: There must be no white space between the comma and the '@' or '!' sign.*

## Data Storage

The instrument's data is stored in two places. A small subset is stored within the RLF file, and the complete set is stored in a separate file on the vehicle: "C:\GENERIC.DAT". This latter file can be downloaded by selecting the menu item "Download | Download data | C:\GENERIC.DAT". The user will be prompted for a destination, which by default is the data directory of the current project. The file name is the mission name with the extension ".GNR". For example, if this is mission 10 in the project test1, then the file name will be:

```
"...\TEST1\DATA\MSN010.GNR"
```

The following data is stored for each message:

```
latitude, longitude,
mission time
depth in meters
altitude in meters
version of the specification (0 for the current revision)
message type
flags
values (up to 8 floating point parameters)
```

The data is stored using REMUS' internal data format and may be exported to either text or Matlab using the REMUS VIP software exporting tools. To export a .GNR file:

1. Load the file into the VIP as if it were a .RLF file
2. Select the menu item "Export | RLF data exporting"
3. From the popup dialog "RLF Data Converter," select "Generic Data."

From there, proceed as you would for exporting other REMUS data types.

The flags value contains the good/bad/abort status of each variable, 2 bits for each value. Each pair of bits starting with the LSB contains either 00 (ok), 01 (bad), or 10 (abort). The value 11 is reserved. Thus the status of all eight parameters is stored within a single 16-bit integer.

There are few subtleties that make exporting this data slightly different than exporting other data. The messages may contain a variable number of values. By default, the exporter assumes that the maximum eight parameters are present, however depending on the message type, there may be fewer. In this case, if the export is in ASCII format, the "missing" parameters are exported as series of comma-separated spaces. If the exported file is a Matlab file, the missing values are set

to 0. The user may manually select a subset of values to be exported, it is not necessary to export all eight parameters.

Different message types naturally will have different meanings for the same parameter position. In order to differentiate the messages, the message type is also exported, however not as a string, but as a long integer generated from the first four characters of the message type. The fifth character is ignored in this conversion. Thus, it is recommended that message types should differ by at least one character of the first four. For example, if message types PMSG1 and PMSG2 are used, both will indicate the same message type when exported because the first four letters are the same (PMSG). For those who are interested, the conversion is done by evaluating the first four characters as a 32-bit (4-byte) integer.

Below is an example of data exported to a text file. The individual lines have been broken into multiple lines to fit the formatting of this document.

```
41.5193215258, -70.6954399025, 9:34:32.2, 16.8558826,
3.17453861, 0,
     1397567824, 0, 123.400002, 213.399994, , , , , , ,
41.5193215258, -70.6954399025, 9:34:32.2, 16.8558826,
3.17453861, 0,
     1397568080, 0, 233.5, , , , , , , ,
41.5193215258, -70.6954399025, 9:34:32.2, 16.8558826,
3.17453861, 0,
     1397568336, 0, 23.3999996, 788.400024, 567, , , , , ,
```

## The User Interface

The user interface on the REMUS VIP contains a number of elements. The first is the "annunciator" (idiot light). This will appear whenever the instrument is enabled on the vehicle. If it is green, the instrument is providing valid serial messages of all types to the vehicle. If any message type stops transmitting for more than 5 seconds, the annunciator will turn red. The annunciator will also turn red if any of the variables transmitted is prefixed by an exclamation point. If a parameter is prefixed by an exclamation point, launch will be disabled, unless "Abort launch is broken" is set to "no," in which case the annunciator will be yellow, and launch will be permitted.

```
Sea Dog Data

Time:        15:26:26.0
Latitude:    41N31.058'
Longitude:   70W41.943'
Depth:       -0.1
Alt:          19.4
P1MSG        Var00: 123.4
             Var01: @213.4
P2MSG        Var10: !233.5
*P3MSG       Var20: 0
             Var21: 788.4
             Var22: 567
```

The DLL will add a text page to the REMUS display. The appearance is as shown above. There are three main columns. The left column contains the data type, the center column contains the variable name, and the right column contains the value of the variable. Since there may be several different

message types, an asterisk is used to show which type of message was most recently received. For display and logging within the .RLF file, the data is down sampled to one of each type of message every 1.0 seconds (thus there may be as many as 12 messages per second). The full data set is recorded in the GENERIC.DAT file on the vehicle. The middle column is the variable name, as derived from the information in the vehicle .INI file. The right-hand column has the most recent value for that variable. This value will be preceded by an exclamation point ('!') or an "at" sign (@) if they had been present in the message from the instrument. In the above display, Var01 has an @ in front of it, indicating that the abort flag was set for this variable. The Var10 has an exclamation point in front of it, indicating that the value is no good, and that launch should not be allowed.

The user interface will also add entries to the plotter ("Export | Plotter...") for each variable. This allows both spatial and linear plots of the data collected.

# APPENDIX B: SENSOR MODULE SHELL AND PRELIMINARY PRESSURE VESSEL DESIGN

### Basic Sensor Module Shell

Note: if the shell will not be used as an integral part of the pressure vessel, holes should be drilled for flooding/draining.

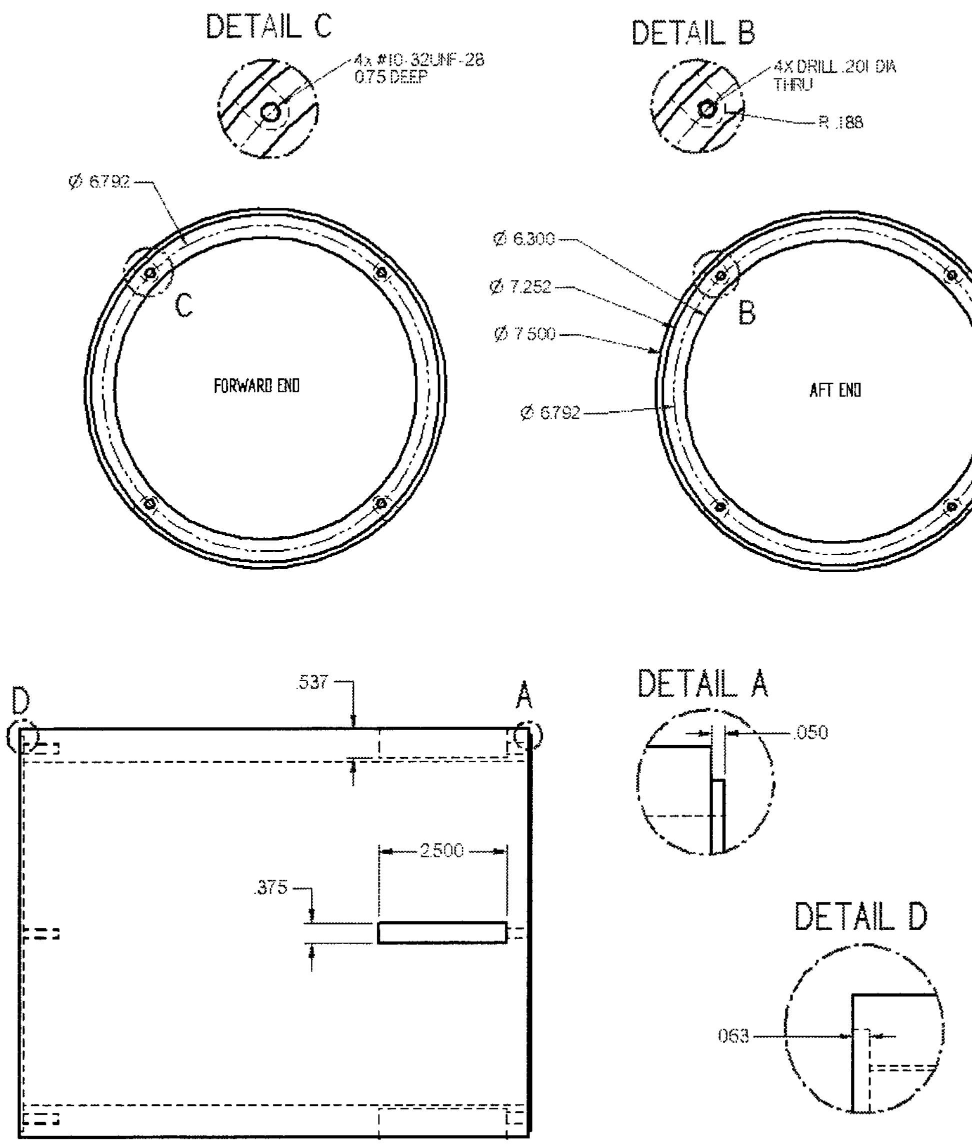



Figure B-1. Basic sensor module shell.

## Preliminary Pressure Vessel Design

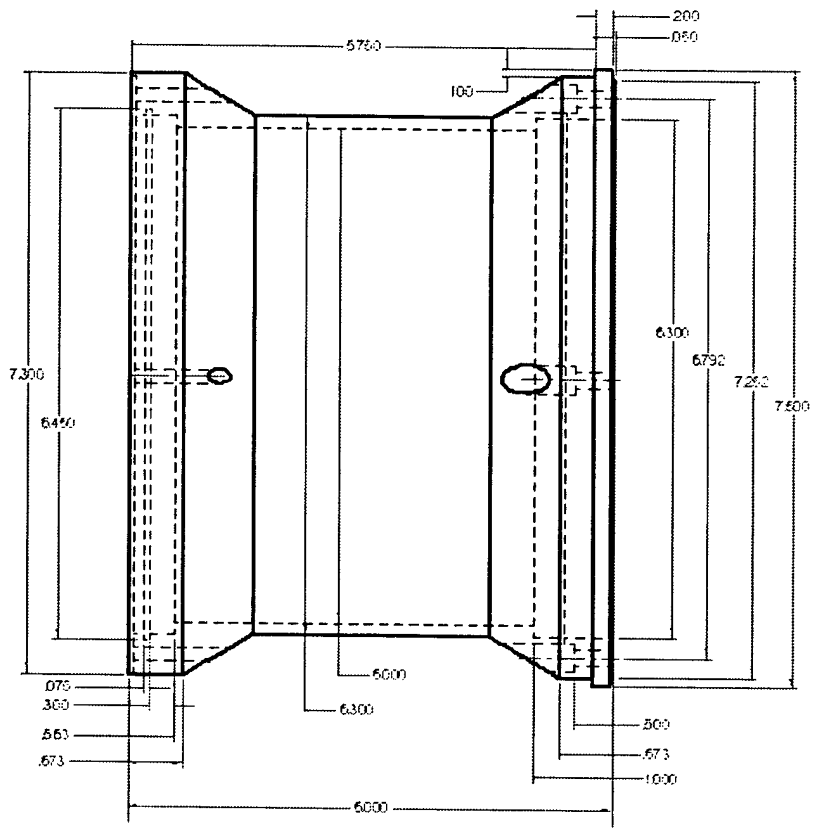



Figure B-2. Pressure vessel side view.

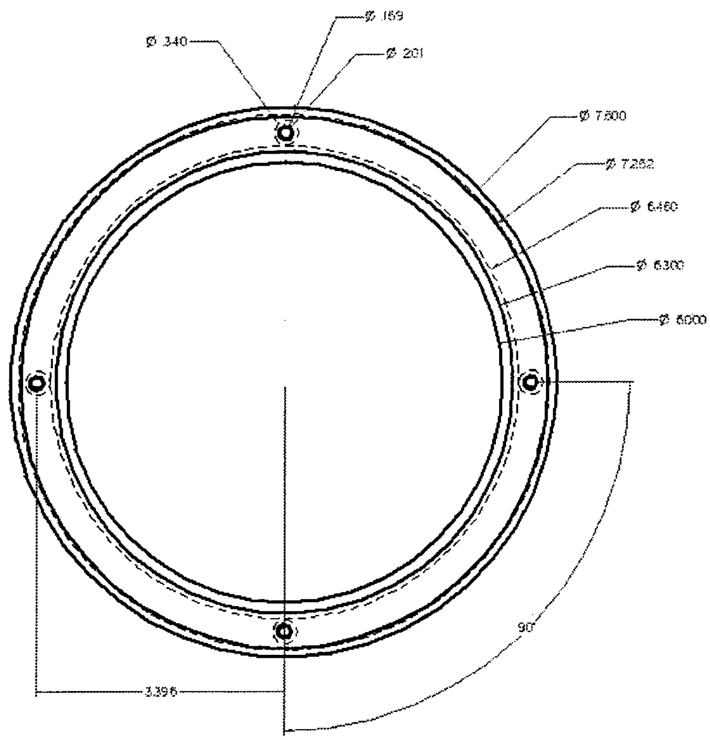



Figure B-3. PV/collar interface.

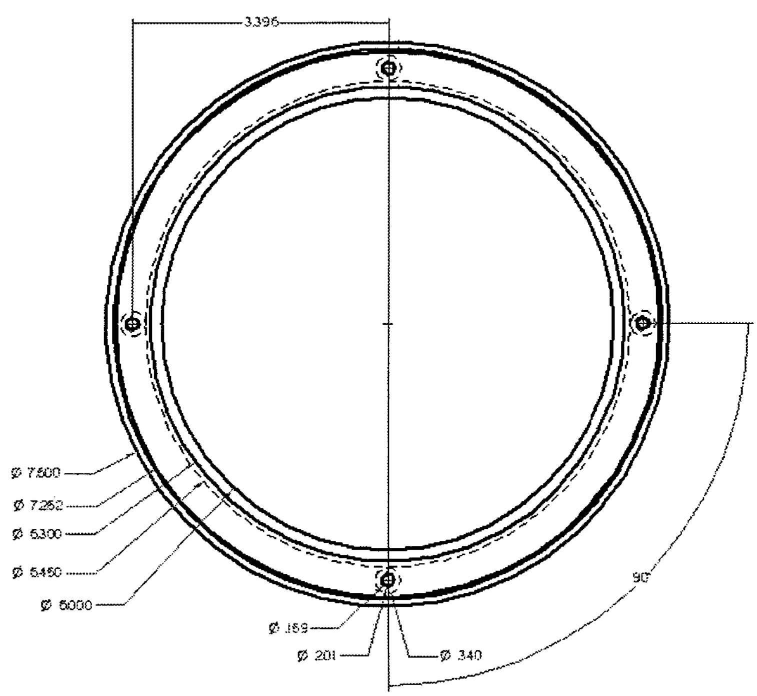



Figure B-4. PV/nose interface.

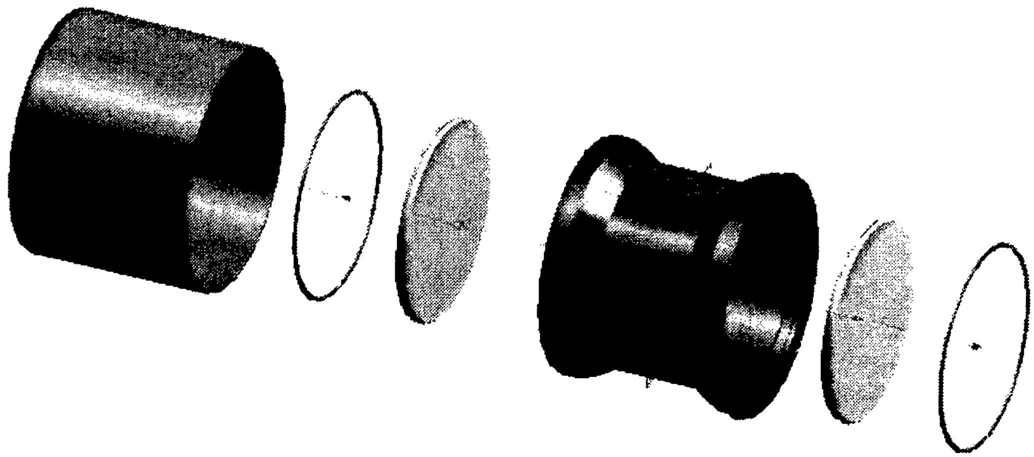


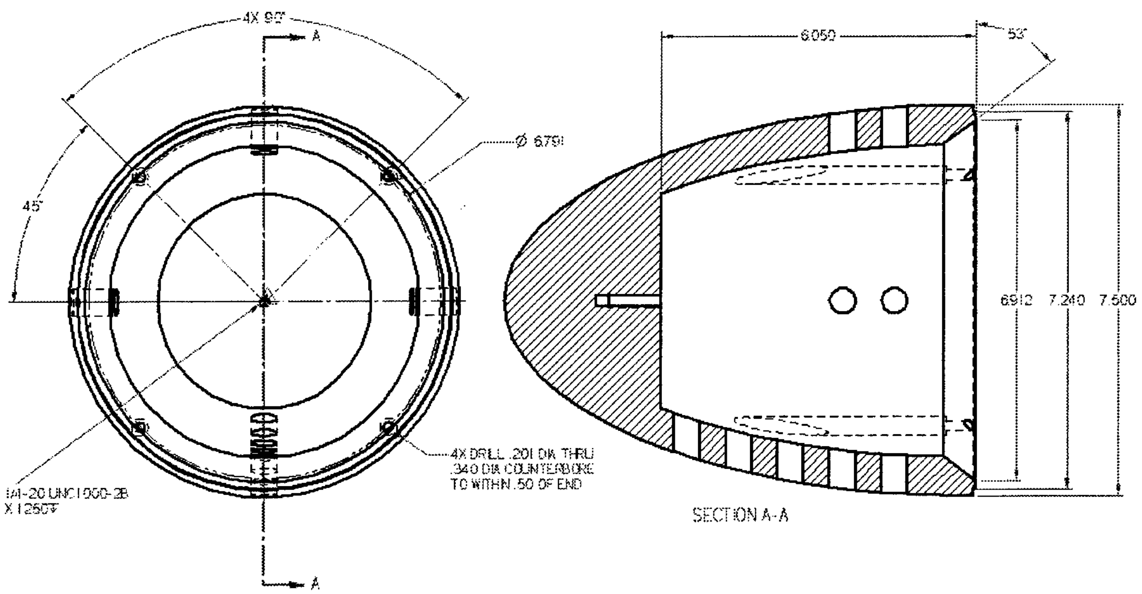
Figure B-5. PV shown with conceptual sleeve (nose interface on the right).

Notes: While the PV/collar and PV/nosecone interfaces look very similar; the PV/nosecone interface will have threaded holes for fasteners from the nosecone, while the PV/collar interface will have holes to insert fasteners for connection to the collar. The pressure vessel will also be designed with a lip to potentially accommodate a sleeve that encompasses the entire body for hydrodynamic purposes.

# APPENDIX C: “DUMMY” NOSECONE

### Drawing of “Dummy” Nosecone

Internal space is available for adding weight or floatation as necessary.



| REPORT DOCUMENTATION PAGE | | | | | | Form Approved OMB No. 0704-01-0188 |
|---|---|---|---|---|---|---|



| **1. REPORT DATE** *(DD-MM-YYYY)* | **2. REPORT TYPE** | **3. DATES COVERED** *(From - To)* |
|---|---|---|
| 02–2004 | Final | 30-10-2002 – 30-09-2004 |

| | |
|---|---|
| **4. TITLE AND SUBTITLE**<br>SSC SAN DIEGO REMUS UUV<br>Sensor Integration | **5a. CONTRACT NUMBER**<br>N0001403WR20113<br>**5b. GRANT NUMBER**<br><br>**5c. PROGRAM ELEMENT NUMBER** |
| **6. AUTHORS**<br>Rich Arrieta, Brian Granger, Vladimir Djapic, Andy Dreiling | **5d. PROJECT NUMBER**<br><br>**5e. TASK NUMBER**<br><br>**5f. WORK UNIT NUMBER** |
| **7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)**<br>SSC San Diego<br>San Diego, CA 92152–5001 | **8. PERFORMING ORGANIZATION REPORT NUMBER**<br>TD 3176 |
| **9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)**<br>Office of Naval Research<br>POC: Dr Keith Ward – ONR342<br>Ballston Tower 1<br>800 N Quincy St, Arlington, VA 22217-5660 | **10. SPONSOR/MONITOR'S ACRONYM(S)**<br>ONR<br>**11. SPONSOR/MONITOR'S REPORT NUMBER(S)** |

**12. DISTRIBUTION/AVAILABILITY STATEMENT**

Approved for public release; distribution is unlimited.

**13. SUPPLEMENTARY NOTES**

This is the work of the United States Government and therefore is not copyrighted. This work may be copied and disseminated without restriction. Many SSC San Diego public release documents are available in electronic format at http://www.spawar.navy.mil/sti/publications/pubs/index.html

**14. ABSTRACT**

This document specifies a standard for integrating environmental sensors onto the Space and Naval Warfare Systems Center San Diego (SSC San Diego) Remote Environmental Monitoring Unit Support (REMUS) unmanned underwater vehicle (UUV). If sensor developers construct sensors in accordance with the guidance provided in this document, they can do so without having access to the vehicle itself, and final electrical and mechanical integration of the unit will be a trivial task.

**15. SUBJECT TERMS**

Mission Area: Intelligence, Surveillance, Reconnaissance
REMUS, UUVs, environmental sensors

| **16. SECURITY CLASSIFICATION OF:** | | | **17. LIMITATION OF ABSTRACT** | **18. NUMBER OF PAGES** | **19a. NAME OF RESPONSIBLE PERSON** |
|---|---|---|---|---|---|
| **a. REPORT** | **b. ABSTRACT** | **c. THIS PAGE** | | | Rich Arrieta |
| | | | | | **19B. TELEPHONE NUMBER** *(Include area code)* |
| U | U | U | UU | 38 | (619) 553–1968 |

## INITIAL DISTRIBUTION

| | | |
|---|---|---|
| 20012 | Patent Counsel | (1) |
| 20271 | Archive/Stock | (1) |
| 20274 | Library | (1) |
| 2027 | M. E. Cathcart | (1) |
| 20271 | F. F. Roessler | (1) |
| 2744 | R. Arrieta | (30) |

Defense Technical Information Center
Fort Belvoir, VA 22060–6218 (4)

SSC San Diego Liaison Office
C/O PEO-SCS
Arlington, VA 22202–4804 (1)

Center for Naval Analyses
Alexandria, VA 22311–1850 (1)

Office of Naval Research
ATTN: NARDIC
Philadelphia, PA 19111-5078 (1)

Government-Industry Data Exchange
Program Operations Center
Corona, CA 91718–8000 (1)